

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-056651
 (43)Date of publication of application : 22.02.2002

(51)Int.Cl. G11B 27/00
 G11B 20/12
 H04N 5/85
 H04N 5/91

(21)Application number : 2001-109342 (71)Applicant : SONY CORP
 (22)Date of filing : 06.04.2001 (72)Inventor : KATO MOTOKI
 HAMADA TOSHIYA

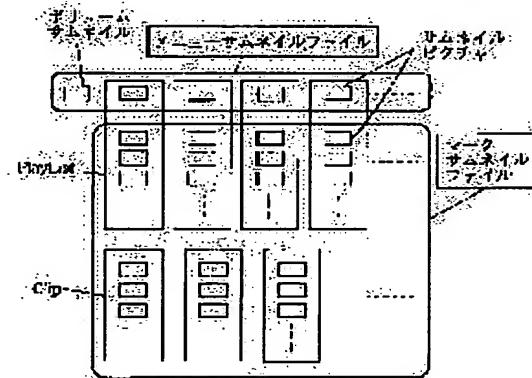
(30)Priority
 Priority number : 2000185479 Priority date : 21.04.2000 Priority country : JP

(54) RECORDER AND RECORDING METHOD, REPRODUCER AND REPRODUCING METHOD, RECORDING MEDIUM AND PROGRAM

(57)Abstract:

PROBLEM TO BE SOLVED: To easily select the recorded data.

SOLUTION: A volume thumbnail is equal to a thumbnail image that symbolizes all data that are recorded on a recording medium. A play list is one of units of reproduction sections of data and plural thumbnails are included in each unit. Then the thumbnails representing the play lists are registered as menu thumbnails. These menu thumbnails are displayed in a list when the play lists are selected. Every play list and also the clip corresponding to the play list include plural thumbnails respectively. These thumbnails are used as mark thumbnails when the reproduction positions are designated in the play lists.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2002-56651

(P2002-56651A)

(43)公開日 平成14年2月22日 (2002.2.22)

(51)Int.Cl'

G 11 B 27/00
20/12

識別記号

1 0 3

H 04 N 5/85
5/91

F I

G 11 B 27/00
20/12

テマコト*(参考)

D 5 C 0 5 2
5 C 0 5 3

1 0 3 5 D 0 4 4

H 04 N 5/85
5/91

B 5 D 1 1 0
N

審査請求 未請求 請求項の数16 OL (全37頁)

(21)出願番号

特願2001-109342(P2001-109342)

(22)出願日

平成13年4月6日 (2001.4.6)

(31)優先権主張番号

特願2000-185479(P2000-185479)

(32)優先日

平成12年4月21日 (2000.4.21)

(33)優先権主張国

日本 (J P)

(71)出願人

000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72)発明者

加藤 元樹

東京都品川区北品川6丁目7番35号 ソニ

ー株式会社内

(72)発明者

浜田 俊也

東京都品川区北品川6丁目7番35号 ソニ

ー株式会社内

(74)代理人

100082131

弁理士 稲本 義雄

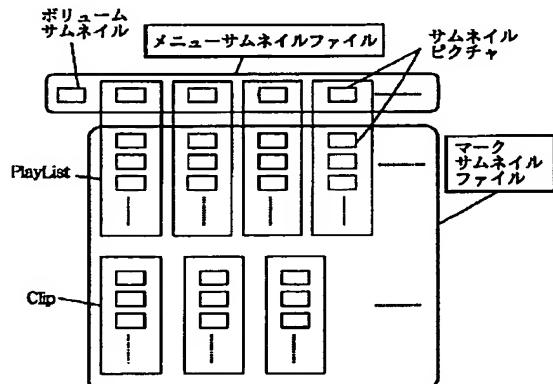
最終頁に続く

(54)【発明の名称】 記録装置および方法、再生装置および方法、記録媒体、プログラム、並びに記録媒体

(57)【要約】

【課題】 記録されているデータの選択を簡便に行えるようにする。

【解決手段】 ボリュームサムネイルは、記録媒体に記録されている全てのデータを象徴するようなサムネイル画像である。プレイリストは、データの再生区間の1つの単位であり、その単位内には、複数のサムネイルが設けられている。そのうち、プレイリストを代表するサムネイルがメニュー・サムネイルとして登録される。このメニュー・サムネイルは、プレイリストを選択する際に、一覧表示される。プレイリストや、そのプレイリストに対応するクリップ内にも、複数のサムネイルが設けられており、それらのサムネイルは、マークサムネイルとして、プレイリスト内の再生位置を指定する際に用いられる。



【特許請求の範囲】

【請求項 1】 入力された動画像データから、当該動画像データを代表するサムネイル画像を第 1 のサムネイルデータとして生成すると共に、前記動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、または、ユーザが指定した画像のサムネイル画像を第 2 のサムネイルデータとして生成する生成手段と、前記生成手段により生成された前記第 1 のサムネイルデータと前記第 2 のサムネイルデータを、各々独立したグループとして記録媒体に記録する記録手段とを有することを特徴とする記録装置。

【請求項 2】 前記生成手段は、前記第 1 のサムネイルデータと前記第 2 のサムネイルデータを、各々独立したファイルとして生成することを特徴とする請求項 1 に記載の記録装置。

【請求項 3】 前記生成手段は、前記第 1 のサムネイルデータと前記第 2 のサムネイルデータを各々独立したデータブロックとするとともに 1 つのファイルとして生成することを特徴とする請求項 1 に記載の記録装置。

【請求項 4】 前記生成手段は、前記第 1 のサムネイルデータに対応する番号で前記第 1 のサムネイルデータを管理する第 1 の管理データも生成すると共に、前記第 2 のサムネイルデータに対応する番号で前記第 2 のサムネイルデータを管理する第 2 の管理データも生成し、前記記録手段は、前記第 1 の管理データおよび前記第 2 の管理データを前記記録媒体に記録することを特徴とする請求項 1 に記載の記録装置。

【請求項 5】 前記第 1 の管理データと前記第 2 の管理データは、管理する前記サムネイル画像の画像データのフォーマット形式を示すデータを含むことを特徴とする請求項 4 に記載の記録装置。

【請求項 6】 前記記録手段は、前記第 1 のサムネイルデータまたは前記第 2 のサムネイルデータに含まれる前記サムネイル画像の画像データを、所定の大きさのブロック単位で記録することを特徴とする請求項 4 に記載の記録装置。

【請求項 7】 前記記録手段は、前記第 1 のサムネイル画像の参照先を示す情報をさらに別ファイルとして前記記録媒体に記録することを特徴とする請求項 1 に記載の記録装置。

【請求項 8】 前記記録手段は、前記第 2 のサムネイルデータに含まれる前記サムネイル画像の参照先を示す情報をさらに記録することを特徴とする請求項 1 に記載の記録装置。

【請求項 9】 入力された動画像データから、当該動画像データを代表するサムネイル画像を第 1 のサムネイルデータとして生成すると共に、前記動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、または、ユーザが指定した画像のサムネイル画像を第 2 のサムネイルデータとして生成する生成ステップ

と、

前記生成ステップの処理で生成された前記第 1 のサムネイルデータと前記第 2 のサムネイルデータを、各々独立したグループとして記録媒体に記録するための制御を行う記録制御ステップとを含むことを特徴とする記録方法。

【請求項 10】 入力された動画像データから、当該動画像データを代表するサムネイル画像を第 1 のサムネイルデータとして生成すると共に、前記動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、または、ユーザが指定した画像のサムネイル画像を第 2 のサムネイルデータとして生成する生成ステップと、

前記生成ステップの処理で生成された前記第 1 のサムネイルデータと前記第 2 のサムネイルデータを、各々独立したグループとして記録媒体に記録するための制御を行う記録制御ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項 11】 入力された動画像データから、当該動画像データを代表するサムネイル画像を第 1 のサムネイルデータとして生成すると共に、前記動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、または、ユーザが指定した画像のサムネイル画像を第 2 のサムネイルデータとして生成する生成ステップと、

前記生成ステップの処理で生成された前記第 1 のサムネイルデータと前記第 2 のサムネイルデータを、各々独立したグループとして記録媒体に記録するための制御を行う記録制御ステップとをコンピュータに実行させるプログラム。

【請求項 12】 画像データの再生が指示された場合、前記画像データの内容を示すサムネイル画像の画像データを管理する管理データを読み出す第 1 の読み出し手段と、

前記読み出し手段により読み出された前記管理データに基づき、前記画像データを読み出す第 2 の読み出し手段と、

前記画像データから、特徴的な画像として抽出された画像のサムネイル画像の画像データか、または、ユーザにより指定された画像のサムネイル画像の画像データと、その画像データを管理する管理データを読み出す第 3 の読み出し手段と、

前記第 3 の読み出し手段により読み出された前記管理データに基づき、前記画像データを読み出す第 4 の読み出し手段とを含むことを特徴とする再生装置。

【請求項 13】 画像データの再生が指示された場合、前記画像データの内容を示すサムネイル画像の画像データを管理する管理データの読み出しを制御する第 1 の読み出し制御ステップと、

前記読み出し制御ステップの処理で読み出しが制御された前記管理データに基づき、前記画像データの読み出しを制御する第2の読み出し制御ステップと、
前記画像データから、特徴的な画像として抽出された画像のサムネイル画像の画像データか、または、ユーザにより指定された画像のサムネイル画像の画像データと、
その画像データを管理する管理データを読み出す第3の読み出しステップと、
前記第3の読み出し制御ステップの処理で読み出しが制御された前記管理データに基づき、前記画像データを読み出す第4の読み出しステップとを含むことを特徴とする再生方法。

【請求項14】 画像データの再生が指示された場合、前記画像データの内容を示すサムネイル画像の画像データを管理する管理データの読み出しを制御する第1の読み出し制御ステップと、
前記読み出し制御ステップの処理で読み出しが制御された前記管理データに基づき、前記画像データの読み出しを制御する第2の読み出し制御ステップと、
前記画像データから、特徴的な画像として抽出された画像のサムネイル画像の画像データか、または、ユーザにより指定された画像のサムネイル画像の画像データと、
その画像データを管理する管理データを読み出す第3の読み出しステップと、
前記第3の読み出し制御ステップの処理で読み出しが制御された前記管理データに基づき、前記画像データを読み出す第4の読み出しステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項15】 画像データの再生が指示された場合、前記画像データの内容を示すサムネイル画像の画像データを管理する管理データの読み出しを制御する第1の読み出し制御ステップと、
前記読み出し制御ステップの処理で読み出しが制御された前記管理データに基づき、前記画像データの読み出しを制御する第2の読み出し制御ステップと、
前記画像データから、特徴的な画像として抽出された画像のサムネイル画像の画像データか、または、ユーザにより指定された画像のサムネイル画像の画像データと、
その画像データを管理する管理データを読み出す第3の読み出しステップと、
前記第3の読み出し制御ステップの処理で読み出しが制御された前記管理データに基づき、前記画像データを読み出す第4の読み出しステップとを含むことを特徴とするプログラム。

【請求項16】 画像データ、前記画像データから、前記画像データの内容を示すサムネイル画像の画像データと、そのサムネイル画像のデータを管理する管理データから構成される第1のデータ、および、前記画像データから、特徴的な画像として抽出された画像のサムネイル

画像の画像データか、または、ユーザが指定した画像のサムネイル画像の画像データと、その画像データを管理する管理データから構成される第2のデータが記録されていることを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は記録装置および方法、再生装置および方法、記録媒体、プログラム、並びに記録媒体に関し、特に、データを代表するサムネイルをデータに付加する記録装置および方法、再生装置および方法、記録媒体、プログラム、並びに記録媒体に関する。

【0002】

【従来の技術】 近年、記録再生装置から取り外し可能なディスク型の記録媒体として、各種の光ディスクが提案されつつある。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、ビデオ信号等のAV(Audio Visual)信号を記録するメディアとしての期待が高い。この記録可能な光ディスクに記録するデジタルのAV信号のソース(供給源)としては、CSデジタル衛星放送やBSデジタル放送があり、また、将来はデジタル方式の地上波テレビジョン放送等も提案されている。

【0003】 ここで、これらのソースから供給されるデジタルビデオ信号は、通常MPEG(Moving Picture Experts Group)2方式で画像圧縮されているのが一般的である。また、記録装置には、その装置固有の記録レートが定められている。従来の民生用映像蓄積メディアで、デジタル放送由来のデジタルビデオ信号を記録する場合、アナログ記録方式であれば、デジタルビデオ信号をデコード後、帯域制限をして記録する。あるいは、MPEG1 Video、MPEG2 Video、DV方式をはじめとするデジタル記録方式であれば、1度デコードされた後に、その装置固有の記録レート・符号化方式で再エンコードされて記録される。

【0004】 しかしながら、このような記録方法は、供給されたビットストリームを1度デコードし、その後で帯域制限や再エンコードを行って記録するため、画質の劣化を伴う。画像圧縮されたデジタル信号の記録をする場合、入力されたデジタル信号の伝送レートが記録再生装置の記録レートを超えない場合には、供給されたビットストリームをデコードや再エンコードすることなく、そのまま記録する方法が最も画質の劣化が少ない。ただし、画像圧縮されたデジタル信号の伝送レートが記録媒体としてのディスクの記録レートを超える場合には、記録再生装置でデコード後、伝送レートがディスクの記録レートの上限以下になるように、再エンコードをして記録する必要はある。

【0005】 また、入力デジタル信号のビットレートが時間により増減する可変レート方式によって伝送されて

いる場合には、回転ヘッドが固定回転数であるために記録レートが固定レートになるテープ記録方式に比べ、1度バッファにデータを蓄積し、バースト的に記録ができるディスク記録装置が記録媒体の容量をより無駄なく利用できる。

【0006】以上のように、デジタル放送が主流となる将来においては、データストリーマのように放送信号をデジタル信号のまま、デコードや再エンコードすることなく記録し、記録媒体としてディスクを使用した記録再生装置が求められると予測される。

【0007】

【発明が解決しようとする課題】上述したような装置により、複数のデータ（例えば、映像データや音声データなどから構成される番組のデータ）が記録されている記録媒体を再生する際、記録媒体に記録されるデータ量が増加するに従い、どの番組を再生するのか、番組中のどのシーンから再生するのかといった、再生させる前の処理が煩雑になるといった課題があった。

【0008】本発明はこのような状況に鑑みてなされたものであり、データを代表するサムネイルをデータに附加することにより、再生させるデータを簡単に選択できるようにすることを目的とする。

【0009】

【課題を解決するための手段】本発明の記録装置は、入力された動画像データから、当該動画像データを代表するサムネイル画像を第1のサムネイルデータとして生成すると共に、動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、または、ユーザが指定した画像のサムネイル画像を第2のサムネイルデータとして生成する生成手段と、生成手段により生成された第1のサムネイルデータと第2のサムネイルデータを、各々独立したグループとして記録媒体に記録する記録手段を有することを特徴とする。

【0010】前記生成手段は、第1のサムネイルデータと第2のサムネイルデータを、各々独立したファイルとして生成するようになることができる。

【0011】前記生成手段は、第1のサムネイルデータと第2のサムネイルデータを各々独立したデータブロックとするとともに1つのファイルとして生成するようになることができる。

【0012】前記生成手段は、第1のサムネイルデータに対応する番号で第1のサムネイルデータを管理する第1の管理データも生成すると共に、第2のサムネイルデータに対応する番号で第2のサムネイルデータを管理する第2の管理データも生成し、前記記録手段は、第1の管理データおよび第2の管理データを記録媒体に記録するようになることができる。

【0013】前記第1の管理データと第2の管理データは、管理するサムネイル画像の画像データのフォーマット形式を示すデータを含むようにすることができる。

【0014】前記記録手段は、第1のサムネイルデータまたは第2のサムネイルデータに含まれるサムネイル画像の画像データを、所定の大きさのブロック単位で記録するようになることができる。

【0015】前記記録手段は、第1のサムネイル画像の参照先を示す情報をさらに別ファイルとして記録媒体に記録するようになることができる。

【0016】前記記録手段は、第2のサムネイルデータに含まれるサムネイル画像の参照先を示す情報をさらに記録するようになることができる。

【0017】本発明の記録方法は、入力された動画像データから、当該動画像データを代表するサムネイル画像を第1のサムネイルデータとして生成すると共に、動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、または、ユーザが指定した画像のサムネイル画像を第2のサムネイルデータとして生成する生成ステップと、生成ステップの処理で生成された第1のサムネイルデータと第2のサムネイルデータを、各々独立したグループとして記録媒体に記録するための制御を行う記録制御ステップとを含むことを特徴とする。

【0018】本発明の第1の記録媒体のプログラムは、入力された動画像データから、当該動画像データを代表するサムネイル画像を第1のサムネイルデータとして生成すると共に、動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、または、ユーザが指定した画像のサムネイル画像を第2のサムネイルデータとして生成する生成ステップと、生成ステップの処理で生成された第1のサムネイルデータと第2のサムネイルデータを、各々独立したグループとして記録媒体に記録するための制御を行う記録制御ステップとを含むことを特徴とする。

【0019】本発明の第1のプログラムは、入力された動画像データから、当該動画像データを代表するサムネイル画像を第1のサムネイルデータとして生成すると共に、動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、または、ユーザが指定した画像のサムネイル画像を第2のサムネイルデータとして生成する生成ステップと、生成ステップの処理で生成された第1のサムネイルデータと第2のサムネイルデータを、各々独立したグループとして記録媒体に記録するための制御を行う記録制御ステップとを含むことを特徴とする。

【0020】本発明の再生装置は、AVストリームの再生が指示された場合、AVストリームの内容を示すサムネイル画像の画像データを管理する管理データを読み出す第1の読み出し手段と、読み出し手段により読み出された管理データに基づき、画像データを読み出す第2の読み出し手段と、AVストリームから、特徴的な画面として抽出された画面のサムネイル画像の画像データか、または、ユーザにより指定された画面のサムネイル画像の画

像データと、その画像データを管理する管理データを読み出す第3の読み出し手段と、第3の読み出し手段により読み出された管理データに基づき、画像データを読み出す第4の読み出し手段とを含むことを特徴とする。

【0021】本発明の再生方法は、AVストリームの再生が指示された場合、AVストリームの内容を示すサムネイル画像の画像データを管理する管理データの読み出しを制御する第1の読み出し制御ステップと、読み出し制御ステップの処理で読み出しが制御された管理データに基づき、画像データの読み出しを制御する第2の読み出し制御ステップと、AVストリームから、特徴的な画面として抽出された画面のサムネイル画像の画像データか、または、ユーザにより指定された画面のサムネイル画像の画像データと、その画像データを管理する管理データを読み出す第3の読み出しステップと、第3の読み出し制御ステップの処理で読み出しが制御された管理データに基づき、画像データを読み出す第4の読み出しステップとを含むことを特徴とする。

【0022】本発明の第2の記録媒体のプログラムは、AVストリームの再生が指示された場合、AVストリームの内容を示すサムネイル画像の画像データを管理する管理データの読み出しを制御する第1の読み出し制御ステップと、読み出し制御ステップの処理で読み出しが制御された管理データに基づき、画像データの読み出しを制御する第2の読み出し制御ステップと、AVストリームから、特徴的な画面として抽出された画面のサムネイル画像の画像データか、または、ユーザにより指定された画面のサムネイル画像の画像データと、その画像データを管理する管理データを読み出す第3の読み出しステップと、第3の読み出し制御ステップの処理で読み出しが制御された管理データに基づき、画像データを読み出す第4の読み出しステップとを含むことを特徴とする。

【0023】本発明の第2のプログラムは、AVストリームの再生が指示された場合、AVストリームの内容を示すサムネイル画像の画像データを管理する管理データの読み出しを制御する第1の読み出し制御ステップと、読み出し制御ステップの処理で読み出しが制御された管理データに基づき、画像データの読み出しを制御する第2の読み出し制御ステップと、AVストリームから、特徴的な画面として抽出された画面のサムネイル画像の画像データか、または、ユーザにより指定された画面のサムネイル画像の画像データと、その画像データを管理する管理データを読み出す第3の読み出しステップと、第3の読み出し制御ステップの処理で読み出しが制御された管理データに基づき、画像データを読み出す第4の読み出しステップとを含むことを特徴とする。

【0024】本発明の第3の記録媒体は、AVストリーム、AVストリームから、AVストリームの内容を示すサムネイル画像の画像データと、そのサムネイル画像のデータを管理する管理データから構成される第1のデータ、

および、AVストリームから、特徴的な画面として抽出された画面のサムネイル画像の画像データか、または、ユーザが指定した画面のサムネイル画像の画像データと、その画像データを管理する管理データから構成される第2のデータが記録されていることを特徴とする。

【0025】本発明の記録装置および方法、並びに第1のプログラムにおいては、入力された動画像データから、当該動画像データを代表するサムネイル画像が第1のサムネイルデータとして生成されると共に、動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、または、ユーザが指定した画像のサムネイル画像が第2のサムネイルデータとして生成され、第1のサムネイルデータと第2のサムネイルデータが、各自独立したグループとして記録媒体に記録される。

【0026】本発明の再生装置および方法、並びに第2のプログラムにおいては、AVストリームの再生が指示された場合、AVストリームの内容を示すサムネイル画像の画像データを管理する管理データが読み出され、読み出された管理データに基づき、画像データが読み出され、AVストリームから、特徴的な画面として抽出された画面のサムネイル画像の画像データか、または、ユーザにより指定された画面のサムネイル画像の画像データと、その画像データを管理する管理データが読み出され、読み出された管理データに基づき、画像データが読み出される。

【0027】

【発明の実施の形態】以下に、本発明の実施の形態について、図面を参照して説明する。図1は、本発明を適用した記録再生装置1の内部構成例を示す図である。まず、外部から入力された信号を記録媒体に記録する動作を行う部分の構成について説明する。記録再生装置1は、アナログデータ、または、デジタルデータを入力し、記録することができる構成とされている。

【0028】端子11には、アナログのビデオ信号が、端子12には、アナログのオーディオ信号が、それぞれ入力される。端子11に入力されたビデオ信号は、解析部14とAVエンコーダ15に、それぞれ出力される。端子12に入力されたオーディオ信号は、AVエンコーダ15にのみ出力される。解析部14は、入力されたビデオ信号からシーンチェンジなどの特徴点を抽出する。

【0029】AVエンコーダ15は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム(V)、符号化オーディオストリーム(A)、およびAV同期等のシステム情報(S)をマルチブレクサ16に出力する。

【0030】符号化ビデオストリームは、例えば、MPEG (Moving Picture Expert Group) 2方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、MPEG 1方式により符号化されたオーディオストリームや、ドルビーアクション3方式により符号化さ

れたオーディオストリーム等である。マルチブレクサ16は、入力されたビデオおよびオーディオのストリームを、入力システム情報に基づいて多重化して、スイッチ17を介して多重化ストリーム解析部18とソースパケットタイザ19に出力する。

【0031】多重化ストリームは、例えば、MPEG2トランスポートストリームやMPEG2プログラムストリームである。ソースパケットタイザ19は、入力された多重化ストリームを、そのストリームを記録させる記録媒体100のアプリケーションフォーマットに従って、ソースパケットから構成されるAVストリームを符号化する。AVストリームは、ECC(誤り訂正)符号化部20、変調部21で所定の処理が施され、書き込み部22に出力される。書き込み部22は、制御部23から出力される制御信号に基づいて、記録媒体100にAVストリームファイルを書き込む(記録する)。

【0032】デジタルインターフェースまたはデジタルテレビジョンチューナーから入力されるデジタルテレビジョン放送等のトランスポートストリームは、端子13に入力される。端子13に入力されたトランスポートストリームの記録方式には、2通りあり、それらは、トランスペアレントに記録する方式と、記録ビットレートを下げるなどの目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインターフェースとしての端子24から制御部23へ入力される。

【0033】入力トランスポートストリームをトランスペアレントに記録する場合、端子13に入力されたトランスポートストリームは、多重化ストリーム解析部18と、ソースパケットタイザ19に出力される。これ以降の記録媒体100へAVストリームが記録されるまでの処理は、上述の入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【0034】入力トランスポートストリームを再エンコードした後に記録する場合、端子13に入力されたトランスポートストリームは、デマルチブレクサ26に入力される。デマルチブレクサ26は、入力されたトランスポートストリームに対してデマルチブレクス処理を施し、ビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)を抽出する。

【0035】デマルチブレクサ26により抽出されたストリーム(情報)のうち、ビデオストリームはAVデコーダ27に、オーディオストリームとシステム情報はマルチブレクサ16に、それぞれ出力される。AVデコーダ27は、入力されたビデオストリームを復号し、その再生ビデオ信号をAVエンコーダ15に出力する。AVエンコーダ15は、入力ビデオ信号を符号化し、符号化ビデオストリーム(V)をマルチブレクサ16に出力する。

【0036】一方、デマルチブレクサ26から出力され、マルチブレクサ16に入力されたオーディオストリームとシステム情報、および、AVエンコーダ15から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ストリームとして多重化ストリーム解析部18とソースパケットタイザ19にスイッチ17を介して出力される。これ以後の記録媒体100へAVストリームが記録されるまでの処理は、上述の入力オーディオ信号と、ビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【0037】本実施の形態の記録再生装置1は、AVストリームのファイルを記録媒体100に記録すると共に、そのファイルを説明するアプリケーションデータベース情報も記録する。アプリケーションデータベース情報は、制御部23により作成される。制御部23への入力情報は、解析部14からの動画像の特徴情報、多重化ストリーム解析部18からのAVストリームの特徴情報、および端子24から入力されるユーザからの指示情報である。

【0038】解析部14から供給される動画像の特徴情報は、入力動画像信号の中の特徴的な画像に関する情報であり、例えば、プログラムの開始点、シーンチェンジ点、コマーシャル(CM)の開始・終了点などの指定情報(マーク)であり、また、その指定場所の画像のサムネイル画像の情報も含まれる。

【0039】多重化ストリーム解析部18からのAVストリームの特徴情報は、記録されるAVストリームの符号化情報に関する情報であり、例えば、AVストリーム内のIピクチャのアドレス情報、AVストリームの符号化パラメータ、AVストリームの中の符号化パラメータの変化点情報、ビデオストリームの中の特徴的な画像に関する情報(マーク)などである。

【0040】端子24からのユーザの指示情報は、AVストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクター文字、ユーザが好みのシーンにセットするブックマークやリリューム点の情報などである。

【0041】制御部23は、上記の入力情報に基づいて、AVストリームのデータベース(Clip)、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベース、記録媒体100の記録内容の管理情報(info.dvr)、およびサムネイル画像の情報を作成する。これら的情報から構成されるアプリケーションデータベース情報は、AVストリームと同様にして、ECC符号化部20、変調部21で処理されて、書き込み部22へ入力される。書き込み部22は、制御部23から出力される制御信号に基づいて、記録媒体100へデータベースファイルを記録する。

【0042】上述したアプリケーションデータベース情報についての詳細は後述する。

【0043】このようにして記録媒体100に記録されたAVストリームファイル(画像データと音声データのフ

ファイル)と、アプリケーションデータベース情報が再生される場合、まず、制御部23は、読み出し部28に対して、記録媒体100からアプリケーションデータベース情報を読み出すように指示する。そして、読み出し部28は、記録媒体100からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベース情報は、復調部29、ECC復号部30の処理を経て、制御部23へ入力される。

【0044】制御部23は、アプリケーションデータベース情報に基づいて、記録媒体100に記録されているPlayListの一覧を端子24のユーザインターフェースへ出力する。ユーザは、PlayListの一覧から再生したいPlayListを選択し、再生を指定されたPlayListに関する情報が制御部23へ入力される。制御部23は、そのPlayListの再生に必要なAVストリームファイルの読み出しを、読み出し部28に指示する。読み出し部28は、その指示に従い、記録媒体100から対応するAVストリームを読み出し復調部29に出力する。復調部29に入力されたAVストリームは、所定の処理が施されることにより復調され、さらにECC復号部30の処理を経て、ソースデバッケッタイザ31出力される。

【0045】ソースデバッケッタイザ31は、記録媒体100から読み出され、所定の処理が施されたアプリケーションフォーマットのAVストリームを、デマルチブレクサ26に出力できるストリームに変換する。デマルチブレクサ26は、制御部23により指定されたAVストリームの再生区間(PlayItem)を構成するビデオストリーム(V)、オーディオストリーム(A)、およびAV同期等のシステム情報(S)を、AVデコーダ27に出力する。AVデコーダ27は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子32と端子33から出力する。

【0046】また、ユーザインターフェースとしての端子24から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部23は、AVストリームのデータベース(Clip)の内容に基づいて、記録媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを、読み出し部28に指示する。例えば、ユーザにより選択されたPlayListを、所定の時刻から再生する場合、制御部23は、指定された時刻に最も近いタイムスタンプを持つIピクチャからのデータを読み出すように読み出し部28に指示する。

【0047】また、ユーザによって高速再生(Fast-forward playback)が指示された場合、制御部23は、AVストリームのデータベース(Clip)に基づいて、AVストリームの中のI-ピクチャデータを順次連続して読み出すように読み出し部28に指示する。

【0048】読み出し部28は、指定されたランダムアクセスポイントからAVストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生さ

れる。

【0049】次に、ユーザが、記録媒体100に記録されているAVストリームの編集をする場合を説明する。ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合、例えば、番組Aという歌番組から歌手Aの部分を再生し、その後続けて、番組Bという歌番組の歌手Aの部分を再生したいといった再生経路を作成したい場合、ユーザインターフェースとしての端子24から再生区間の開始点(イン点)と終了点(アウト点)の情報が制御部23に入力される。制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成する。

【0050】ユーザが、記録媒体100に記録されているAVストリームの一部を消去したい場合、ユーザインターフェースとしての端子24から消去区間のイン点とアウト点の情報が制御部23に入力される。制御部23は、必要なAVストリーム部分だけを参照するようにPlayListのデータベースを変更する。また、AVストリームの不必要的ストリーム部分を消去するように、書き込み部22に指示する。

【0051】ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合であり、かつ、それぞれの再生区間をシームレスに接続したい場合について説明する。このような場合、制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成し、さらに、再生区間の接続点付近のビデオストリームの部分的な再エンコードと再多重化を行う。

【0052】まず、端子24から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部23へ入力される。制御部23は、読み出し部28にイン点側のピクチャとアウト点側のピクチャを再生するために必要なデータの読み出しを指示する。そして、読み出し部28は、記録媒体100からデータを読み出し、そのデータは、復調部29、ECC復号部30、ソースデバッケッタイザ31を経て、デマルチブレクサ26に出力される。

【0053】制御部23は、デマルチブレクサ26に入力されたデータを解析して、ビデオストリームの再エンコード方法(picture_coding_typeの変更、再エンコードする符号化ビット量の割り当て)と、再多重化方式を決定し、その方式をAVエンコーダ15とマルチブレクサ16に供給する。

【0054】次に、デマルチブレクサ26は、入力されたストリームをビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)に分離する。ビデオストリームは、「AVデコーダ27に入力されるデータ」と「マルチブレクサ16に入力されるデータ」がある。前者のデータは、再エンコードするために必要なデータ

であり、これはAVデコーダ27で復号され、復号されたピクチャはAVエンコーダ15で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチブレクサ16に入力される。

【0055】マルチブレクサ16は、制御部23から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、EC C符号化部20、変調部21で処理されて、書き込み部22に入力される。書き込み部22は、制御部23から供給される制御信号に基づいて、記録媒体100にAVストリームを記録する。

【0056】このような記録再生装置1において記録媒体100に記録されるデータ（記録媒体100から再生されるデータ）について説明する。MPEG Video、MPEG Audio等の符号化方式で符号化され、MPEG-2 Systemsに従って多重化されたビットストリームを、ファイルシステムが扱うファイルの形にしてディスク（以下、適宜、記録媒体100をディスク状の記録媒体とし、ディスクと称する）に記録したものをAV(Audio Video)ストリームファイル（またはClip AVストリーム）と称する。

【0057】このようなAVストリームファイルの一部または全部の範囲を指定して、必要な部分だけを並べて再生する再生順序指定の仕組みを説明する。図2のように、AVストリームファイルの一部または全部の範囲を指定して、必要な部分だけを再生する再生順序指定がPlaylistである。Playlistは、ユーザから見て、ひとまとまりの単位である。最も簡単な構成になるのは、記録開始をしてから記録終了までの単位で、編集をしなければ、これが1つのPlaylistになる。

【0058】Playlistは、どのAVストリームを再生するかという、AVストリームファイルの指定と、そのファイル中の再生開始点（イン点）と再生終了点（アウト点）の集まりで構成される。AVストリームファイル、再生開始点、再生終了点等を1組とし、これをPlayitemと称する。すなわち、PlaylistはPlayitemの集合である。

【0059】図2に示したように、Playitemは、あるAVストリームファイルの、イン点、アウト点で指定する範囲を参照する。Playitemを再生するということは、そのPlayitemが参照するAVストリームの一部分を再生するということになる。

【0060】AVストリームは、MPEG-TSの形などにマルチブレクスされているビットストリームであるが、このAVストリームが記録されるファイルとは別のファイルに、そのビットストリームに対して1対1に対応する情報（クリップ情報：Clip information）を保持する。これは、再生、編集をより容易にするために設けられている。このようなクリップ情報と、AVストリームの両方をひとまとまり（オブジェクト）とみなし、これをClip

（クリップ）と呼ぶ。すなわち、ClipはAVストリームと、それに付随する情報から構成される1つのオブジェクトである。

【0061】以上のような、関係をUML図で表すと、図3のようになる。図3のUML図で表されるAVストリームファイル、Clip、Playitem、Playlistの構造により、AVストリームファイルを変更しない、任意の部分だけを再生する非破壊再生順序指定が可能になる。

【0062】次に、本発明を実現するための、各種の情報が記録または再生される記録媒体（メディア）上のファイル配置について説明する。メディア上には、図4に示すように、info.dvr、menu.thmb（mark.thmb）、#####.rpls（#####.vpls）（#####は任意の番号）、%%%%%.cli（%%%%%は任意の番号）、および%%%%%.m2ts（%%%%%は、各m2tsファイルがcliファイルと1対1に対応するような番号）の、5種類のファイルが記録される。

【0063】ディスク上にディレクトリ/DVRを用意し、このディレクトリの下が、1つのディスク記録再生システムで管理される範囲とする。/DVRは、ディスクのルートディレクトリにあっても良いし、所定のディレクトリの下に存在していても良い。/DVRディレクトリには、info.dvr、menu.thmb、mark.thmbというファイルが置かれる。また、/DVRの下には、/PLAYLIST、/CLIPINF、/M2TSというディレクトリが置かれる。ファイル#####.rpls、#####.vplsは、/PLAYLISTの下に置かれ、/CLIPINFには%%%%%.cliが置かれ、/M2TSには、%%%%%.m2tsが置かれる。

【0064】ファイルinfo.dvrは、/DVRの下にただ1つある。info.dvrの構造は、図5に示したようなシンタクスで表される。ファイルの内部は、機能別に情報ごとにブロックを構成しており、volumeに関する情報はDVRVolume()に、Playlistの並びに関する情報はTableOfPlayLists()に、記録再生装置1のメーカ固有の情報はMakerPrivateData()に、それぞれ格納される。ファイルの先頭部分には、それらのブロックの先頭を表すアドレスが記述されている。例えば、TableOfPlayLists_Start_addressは、TableOfPlayLists()が開始する位置を、ファイル内の相対バイト数で表したものになっている。

【0065】DVRVolume()の構造は図6に示したようなシンタクスで表される。version_numberは、DVRVolume()のバージョン番号を表し、lengthは、length直後のフィールドからDVRVolume()の最後までの長さをバイトで表したものである。ResumeVolume()はresumeに関する情報を格納し、UIAppInfoVolume()はVolumeの属性情報を格納する領域である。

【0066】図7は、UIAppInfoVolume()のシンタクスを表したものである。character_setは、Volume_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法としては、ASCII、Unicode等を選択できる。name_lengthは、Volume_nameフィール

ドの中に示されるボリューム（ディスク）名のバイト長を示す。Volume_nameは、ボリューム（ディスク）の名前を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはボリューム（ディスク）の名前を示す。

【0067】Volume_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。Volume_protect_flagは、そのボリュームの中のコンテンツをユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが1にセットされている場合、ユーザが正しくPIN番号（図7中のPIN）を入力できた場合にだけ、ユーザは、そのボリュームのコンテンツを視聴する事ができる。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくとも、ユーザが、そのボリュームのコンテンツを視聴する事ができる。最初に、ユーザが、ディスクをプレーヤ（記録再生装置1）へ挿入した時において、もしこのフラグが0にセットされているか、または、このフラグが1にセットされていてもユーザがPIN番号を正しく入力できたらば、プレーヤーは、そのセットされたディスク中のPlayListの一覧を表示させる。

【0068】以上は、Volumeに対しての再生制限であるが、それぞれのPlayListの再生制限については、Volume_protect_flagとは無関係であり、それは後述する図9のUIAppInfoPlayList()の中で定義されるplayback_control_flagによって示される。PINは、4個の0乃至9までの数字であり、それぞれの数字は、ISO/IEC 646に従って符号化される。ref_thumbnail_indexは、Volumeを代表するサムネイルが存在する場合、そのサムネイルを特定するための番号を格納する領域である。ref_thumbnail_indexで指定されるthumbnail_indexを持つ、ファイルmenu.thmb中のサムネイルが、Volumeを代表するメニュー（本発明では、VolumeやPlayListを代表するサムネイルを特にメニューと称する）となる。

【0069】rp_info_valid_flagは、これが1である場合に次に続くrp_ref_to_PlayList_file_name, rp_ref_to_PlayItem_idおよびrp_time_stampが有効な値を持つことを示す。rp_ref_to_PlayList_file_nameは、上記のVolumeを代表するメニューとサムネイルが、所定のPlayList中の画像から抜き出された画像から作られていることを示し、そのPlayListファイルの名前を示す。

【0070】rp_ref_to_PlayItem_idは、rp_ref_to_PlayList_file_nameで示されるPlayList中の1つのPlayItemを指すPlayItem_idを示し、上記のVolumeを代表するメニューとサムネイルが、そのPlayItem中の画像から抜き出された画像から作られていることを示す。rp_time_stampは、rp_ref_to_PlayItem_idが指すPlayItem中の1つの画像のプレゼンテーションタイムスタンプを示し、その画像から上記のVolumeを代表するメニューとサムネイル

が作られていることを示す。

【0071】図8は、TableOfPlayLists()のシンタクスを表したものである。ここでnumber_of_PlayListsはVolume中のPlayListの数を表し、PlayList_file_nameは####.rpls、#####.vpls等のファイル名を指定するものである。UIAppInfoPlayList()には、PlayListの各種属性がかかるており、シンタクスは図9のようになっている。図9に示したUIAppInfoPlayList()には、PlayListの再生には直接必要ではない、PlayListの名前、記録日時、記録時間、消去禁止の有無等の、各種属性情報が格納される。その中のref_thumbnail_indexにより、PlayListの代表画としてのサムネイルを指定することができる。すなわち、ref_thumbnail_indexで指定されるthumbnail_indexを持つ、ファイルmenu.thmb中のサムネイルが、このPlayListを代表するメニューとサムネイルとなる。

【0072】rp_info_valid_flagは、これが1である場合に、次に続くrp_ref_to_PlayItem_idおよびrp_time_stampが有効な値を持つことを示す。rp_ref_to_PlayItem_idは、PlayListの中の1つのPlayItemを指すPlayItem_idを示し、PlayListを代表するメニューとサムネイルが、そのPlayItem中の画像から抜き出された画像から作られていることを示す。rp_time_stampは、rp_ref_to_PlayItem_idが指すPlayItem中の1つの画像のプレゼンテーションタイムスタンプを示し、その画像からPlayListを代表するメニューとサムネイルが作られていることを示す。

【0073】図4に示したファイルXXXXXX.clpiは、/CLI PINFの下に、各AVストリームファイルXXXXXX.m2tsに対応して1つ作られる。XXXXXX.clpiの構造は、図10に示すようになっている。ファイルの内部は、機能別の情報ごとにブロックを構成しており、Clipに関する情報はClipInfo()に、MPEG-2 systemsにおけるトランスポートストリームの時刻基準を表すPCR（プログラムクロッククリフアレンス）の不連続点に関する情報はSTC_Info()に、MPEG-2 systemsのprogram（プログラム）に関する情報はProgramInfo()に、AVストリーム中のランダムアクセス開始可能点等の特徴的な点を表すCPI（Characteristic Point Information）に関する情報はCPI()に、Clipにつけられた、頭出しのためのインデックス点やコマーシャルの開始・終了点などのマーク情報はClipMark()に、それ格納される。ファイルの先頭部分には、それらのブロックの先頭を表すアドレスが記述されている。

【0074】ここでは、サムネイルに関係するブロックだけを説明する。本実施の形態においては、クリップにサムネイルを付ける操作は、クリップのマークにサムネイルを付与することで実現する。クリップに付けられるマークの情報は、ClipMark()に格納されている。図11に、ClipMark()のシンタクスを示す。シンタクス中のmark_typeが、resume、bookmark、skip等のマークの種類を表し、mark_time_stampでマークがつけられた時

刻を表す。サムネイルに関するフィールドは、ref_thumbnail_indexである。ref_thumbnail_indexは、ここでサムネイルの番号を指定することで、サムネイルのデータが格納されているmark.thumbnailファイル中のサムネイルを特定する。ref_thumbnail_indexで指定されるサムネイルが、このマークに付与されたマークサムネイル（本発明では、マークに付与されたサムネイルを特にマークサムネイルと称する）となる。

【0075】マークは、主としてClipおよびPlaylistの中のハイライトや特徴的なシーンを指示するためにある。また、マークより後の部分は再生を飛ばして、次のPlaylistの再生をするように指示するスキップ機能も、マークによって実現できる。図12は、マークがClip、Playlistに付けられている様子を図で示したものである。以下にマークの特徴をまとめると。

【0076】Clipに付加されるマークは、AVストリームの内容に起因する特徴的なシーン、例えば、シーンチェンジ点などを指定する。Playlistを再生する時には、そのPlaylistが参照しているClip中のマークを利用してランダムアクセス等ができる。図12では、Clipに対してcommercial (CM) とsceneという、2種類のマークが付けられているが、commercialのマークは左のPlaylistから、sceneマークは二つのPlaylistから利用されている。Playlistに付加されるマークは、主としてユーザによって設定される。例えば、ブックマークやリジューム点などである。図12では、bookmarkとresumeマークが相当する。

【0077】ClipまたはPlaylistにマークを設定することは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中からそのマークのタイムスタンプを除去する事である。それゆえ、マークの設定や削除は、AVストリームを何も変化させない。

【0078】次に、図4に示したファイル#####.rpls、#####.vplsについて説明するに、ファイル#####.rpls、#####.vplsは、/PLAYLISTの下に、各playlistに対してどちらか1つ作られる。#####.rpls、#####.vplsの構造は、図13に示すようになっている。ファイルの内部は、機能別的情報ごとにブロックを構成しており、Playlistに関する情報はPlayList()に、Playlistに付けられるマークの情報はPlayListMark()に、このPlaylistファイルを記録した記録再生装置のメーカ固有の情報はMakePrivateData()に、それぞれ格納される。ファイルの先頭部分には、ブロックの先頭を表すアドレス(PlayListMark_Start_address等)が記述されている。これにより、ブロックの前あるいは後ろにpadding_byteを挿入することが出来る。ただし、PlayList()の開始位置は固定で、ファイルの先頭から、例えば、256バイト目と設定されている。

【0079】ブロックPlayList()の内容は、図14のよ

うになっている。最初にversion_numberがあり、以下に続く情報のバージョン番号を表す。lengthは、lengthの直後のフィールドから、PlayList()の終わりまでのバイト長を表す。PlayList_typeは、このPlaylistの種類を表し、CPI_typeは、このPlaylistが持つCPIの種類を表す。number_of_PlayItemsは、このPlaylistを構成するPlayItemの数を表す。number_of_SubPlayItemsは、このPlaylistに付けられているアフレコオーディオ用のPlayItem(SubPlayitem)の数を表す。PlayItem()は、Playitemの情報を、SubPlayItem()は、SubPlayitemの情報を格納する。

【0080】ブロックPlayItem()の内容は、図15のようになっている。Clip_Information_file_nameは、このPlayItemが参照しているClip情報ファイル（拡張子がcli piであるファイル）のファイル名が文字列で格納されている。STC_sequence_idは、program中に存在する、PCRが連続な時間範囲の区間をあらわす。この区間内では、一貫した連続時間軸が定義できるようになっているので、PlayItemの開始・終了点を一意に定めることができる。つまり、各PlayItemの開始点と終了点は、同一のSTC_sequenceに存在していなければならない。

【0081】IN_timeは、このPlayItemの開始点の、STC_sequence上のpts (Presentation Time Stamp)を表し、OUT_timeはPlayItemの終了点の、STC_sequence上のptsを表す。connection_conditionは、このPlayitemが次のPlayitemとの間でどのような接続がされているかを表す情報であり、Playitemの間を継ぎ目なく再生できるかどうかの条件を表す。

【0082】BridgeSequnceInfo()は、Playitemの継ぎ目の部分で、本来再生すべきビットストリームとは異なるビットストリームに飛び、それを代わりに再生することで、Playitemの間をシームレスに再生する機能を実現する際に作成されるビットストリーム（ブリッジシーケンス）に関する情報を格納する。program_numberは、このPlayitemが参照しているprogram(MPEG Systemsで定義されている、ビデオ・オーディオ等のエレメンタリストリームのまとめをいう。いわゆるテレビジョン放送のチャンネルに相当するものである)のprogram_numberを表す。

【0083】以上が、Playlist、Playitemを構成するデータ構造の概要である。このようなデータ構造により、AVストリーム中の再生したい部分をIN、OUT点の組で指定したPlayitemの並びでPlaylistを構築し、ユーザが認識するひとまとまりの再生単位を管理することが可能になる。

【0084】本発明の実施の形態では、プレイリスト上の任意の時刻にサムネイルを付ける操作は、プレイリストのマークにサムネイルを付与することで実現している。Playlistに付けられるマークの情報は、PlayListMark()に格納されている。図16は、PlayListMark()のシ

ンタクスを示す図である。シンタクス中のmark_typeが、resume、bookmark、skip等のマークの種類を表し、mark_time_stampでマークがつけられた時刻を表す。ref_thumbnail_indexは、ここでサムネイルの番号を指定することで、サムネイルのデータが格納されているmark.thmbファイル中のサムネイルを特定する。ref_t thumbnail_indexで指定されるサムネイルが、このマークに対応するマークサムネイルとなる。

【0085】次に、サムネイルの詳細について説明する。サムネイルとは、Volume、Playlist、Clipに付随する静止画のことを指す。サムネイルには2種類ある。一つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソルを操作して見たいものを選択するためのメニュー画面で使われる。もう一つは、マークが指しているシーンを表す画像である。

【0086】Volumeと各Playlistは代表画を持つことができるようになる必要がある。Volumeの代表画とは、ディスクをプレーヤに入れた時に、ディスクの内容を表す静止画を最初に表示する場合などに用いることを想定している。Playlistの代表画とは、Playlistを選択するメニュー画面において、Playlistの内容を表すための静止画として用いられる想定している。

【0087】Playlistの代表画の最も簡単な実現方法は、Playlistの最初の画像をサムネイルにすることであるが、必ずしも再生時刻0の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、Playlistのサムネイルとして、任意の画像を決めることが出来るようにしておく。以上2種類のサムネイルをメニュー・サムネイルと称する。メニュー・サムネイルは、頻繁に表示されるため、ディスクから高速に読み出されることが可能である必要がある。この要求を満たすには、全てのメニュー・サムネイルを1つのファイルに格納することが効率的である。必ずしもボリューム内の動画から抜き出したピクチャだけではなく、図17に示すように、パーソナルコンピュータやデジタルスチルカメラから取り込んだ画像でもよい。

【0088】一方、ClipとPlaylistは複数個のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることが出来るようになる必要がある。このようなマーク点を表すピクチャをマークサムネイルと称する。よって、サムネイルの元となるものは、図18や図19に示したように、外部から取り込んだ画像よりも、マーク点の画像を抜き出したものが主となる。メニュー・サムネイルと異なり、マークサムネイルはPlaylistの詳細を表す時に使われるサブメニュー等で使われるため、短いアクセス時間で読み出されなくて良い。

【0089】そのため、サムネイルが必要になる度に、プレイヤーがファイルを開き、ファイルの一部を読み出すことで多少時間がかかるても、問題にはならない。ま

た、ボリューム内に存在するファイル数を減らすために、すべてのマークサムネイルは一つのファイルに格納するのがよい。Playlistはメニュー・サムネイル一つと複数のマークサムネイルを有することができるが、Clipは直接ユーザが選択する必要性がない（通常、Playlist経由で指定する）ため、メニュー・サムネイルを持つことはしない。図20は、以上の関係を示した図である。

【0090】サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は容易に、かつ、高速に実行されなければならない。この理由のため、Thumbnail10はブロック構造を有する。画像のデータはいくつかの部分に分割され、各部分は1つのtn_block()に格納される。1つの画像データは、連続したtn_block()に格納される。tn_block()の列には、使用されていないtn_block()が存在してもよい。1つのサムネイル画像のバイト長は可変である。

【0091】図21は、サムネイルのデータを格納するファイルのシンタクスである。このファイルには、Thumbnail10がただ一つ存在する。図22は、thubnail10のシンタクスを表す。version_numberは、このthubnai10のバージョンナンバーを示す4個のキャラクター文字を表す。lengthは、このlengthフィールドの直後からthumbnai10の最後までのthumbnai10のバイト数を示す32ビットの符号なし整数である。tn_blocks_start_addrは、thumbnai10中の最初のtn_block()の、thumbnai10の先頭からの開始バイトアドレスを示す32ビットの符号なし整数である。

【0092】number_of_thumbnailsは、このthumbnai10に格納されているサムネイル画像の数を示す16ビットの符号なし整数である。tn_block_sizeは、1 tn_block()の大きさをキロバイト単位で表す16ビットの符号なし整数である。例えば、tn_block_sizeが1であるのは、1つのtn_block()のサイズが1024バイトであることを表す。number_of_tn_blocksは、このthumbnai10中のtn_block()の数を表す16ビットの符号なし整数である。

【0093】thumbnai_indexは、このthumbnai_indexフィールドから始まるforループ1回分のサムネイル情報で表されるサムネイルのインデックス番号を表す16ビットの符号なし整数である。thumbnai_indexとして、0xFFFFという値を使用してはならない。thumbnai_indexはref_thumbnai_indexによって参照される。thumbnai_picture_formatは、サムネイル画像のピクチャフォーマットを表す8ビットの符号なし整数で、図23に示したような値をとる。図23中のDCF（Design rule for Camera File System）とPNG（Portable Network Graphics）は”menu.thmb”内でのみ許される。すなわちマークサムネイルは、値”0x00”（MPEG-2 Video I-picture）とならない。

【0094】picture_data_sizeは、符号化されたサム

ネイル画像のバイト長を表す32ビットの符号なし整数である。start_tn_block_numberは、サムネイル画像のデータが始まるtn_block()のtn_block_numberを表す16ビットの符号なし整数である。サムネイル画像データの先頭は、tb_blockの先頭と一致していなければならない。start_tn_block_numberは、0から始まり、tn_blockのforループ中の変数kの値に関係している。x_picture_lengthは、サムネイル画像の水平方向のピクセル数を表す16ビットの符号なし整数である。

【0095】y_picture_lengthは、サムネイル画像の垂直方向のピクセル数を表す16ビットの符号なし整数である。tn_block()は、ピクチャデータを格納する領域である。thumbnail()中のすべてのtn_block()は、tn_block_sizeで定められる同一の大きさを有していなければならない。図24は、画像データがどのようにtn_block()に格納されるかを模式的に表した図である。図24に示したように、各画像データはtn_block()の先頭から始まり、1tn_block()を超える大きさの場合は、連続する次のtn_block()を使用することにより格納される。

【0096】画像データは、可変長であるが、1tn_block()は、固定長である。このように、可変長であるデータを固定長のデータに変換して扱うようにすることにより、新たな画像データの追加や、画像データの削除といった処理に対してブロック単位で対応できるため、アドレスの管理など、簡便に行なうことが可能となる。

【0097】ここで、サムネイル記録についてのディレクトリ・ファイル構造、シンタックスの別案を示す。まず、図4で表されるディレクトリ・ファイル構造の別案として、図25の構造が考えられる。図4では、マークサムネイルを記録するためのファイルが/DVRの下に1つ、mark.thmbだけであるのに対し、図25では、プレイリスト、クリップごとに1対1に対応してファイルxxx.thmb(xxxxxは対応するプレイリストまたはクリップをあらわす、拡張子を除いたファイル名)が作られている。

【0098】例えば、#####.rplsで表されるプレイリストが有するマークサムネイルの画像は、同じディレクトリ内の#####.thmbに格納されることになる。同様に、%%%clpiで表されるクリップが有するマークサムネイルの画像は、同じディレクトリ内の%%%clpi.thmbに格納されることになる。プレイリスト・クリップのマークサムネイルは、ファイルの拡張子を除いた名前の部分が同一の.thmbファイルに記録されるという制約があるだけで、xxxxx.thmb内のシンタックスは、図26のように、menu.thmbやmark.thmb(図21)と同一である。

【0099】また、プレイリストのメニュー・サムネイルについて、図8、図9では、サムネイルを特定するref_thumbnail_indexを、ファイルInfo.dvrのTableOfPlayLists()の中のUIAppInfoPlayList()に置いているが、UIAppInfoPlayList()をInfo.dvrではなく、各プレイリスト

ファイル中に入れる方法もある。このような場合、UIAppInfoPlayList()を、TableOfPlayLists()から、プレイリストファイル中のplaylist()へ移すことになり、図8に示したシンタックスは図27に示すシンタックスのようになり、図14に示したシンタックスは図28に示すシンタックスのようになる。図28中のUIAppInfoPlayList()は、図29に示すシンタックスのようになり、ここにref_thumbnail_indexが入る。

【0100】以上のようなファイル、シンタックスでもサムネイルの記録が可能である。

【0101】次に、図30に示したフローチャートを参照して、マークサムネイルの作成について説明する。ステップS1において、ユーザは、再生したいPlayList(プレイリスト)を選択する。ステップS2において、選択されたPlayListに基づいて、AVストリームの再生が開始される。ステップS3において、ユーザは、再生されているAVストリームを視聴し、マークしたいシーンを探索する。ユーザは、マークしたいシーンが探索できた場合、記録再生装置1に付属するリモートコントローラ(不図示)のマークボタンを操作する。この操作は、再生中に操作されても、一時停止された状態で操作されても良い。

【0102】ステップS3において、ユーザによりマークボタンが操作されると、ステップS4において、ユーザの指示に対応する処理としてマーク位置が決定される。マーク位置が決定されると、ステップS5において、サムネイル画像にする画像の選択が行われる。マーク位置が指示された時点で、制御部23(図1)は、サムネイルを作成するかどうかをユーザに尋ね、作成する場合には、サムネイルとする画像の候補をコマ送り等によりユーザに提供し、ユーザがサムネイル画像を選択するようにしても良い。

【0103】ユーザ、または、制御部23により、サムネイル画像にする画像が選択されると、制御部23は、ステップS6において、AVデコーダ15から画像を取り込み制御部23のRAM(Random Access Memory)(不図示)に転送する。ステップS7において、画像を圧縮するか否かが判断され、画像を圧縮すると判断された場合、制御部23は、ステップS8に進み、画像圧縮を行い、再びデータをRAMに戻す。ステップS7において、画像を圧縮しないと判断された場合、ステップS8の処理はスキップされ、ステップS9に進む。

【0104】ステップS9において、制御部23は、サムネイルの符号化方式、サムネイルを識別するthumbnail_id、サイズ(バイト数)、X、Y方向のピクセル数から構成されるサムネイルのヘッダ情報を作成し、画像データをtn_block()の単位に分割する。ステップS10において、ECC符号化ブロック20にヘッダ情報とtn_block()の形に分割した画像データが転送され、書き込み部22を経て、mark.thmbファイルとして記録媒体100

に書き込まれる。

【0105】次に、メニューサムネイルを作成について、図31に示したフローチャートを参照して説明する。メニューサムネイルはボリュームまたは各プレイリストに対して付けられるので、制御部23は、ステップS21において、ユーザに対して、メニューサムネイルをつける対象を選択させる。ステップS22において、制御部23は、サムネイルを外部から取り込むか、プレイリスト中のあるシーンから取り込むかをユーザに選択させる。

【0106】ステップS22において、サムネイルを外部から取り込むと判断された場合、ステップS23に進み、ユーザが指定した入力端子あるいはファイルから画像データが取り込まれる。取り込みが終了されると、ステップS26に進む。

【0107】一方、ステップS22において、サムネイルを外部から取り込むのではないと判断された場合、ステップS24に進み、制御部23は、まず、ユーザに対して、どのプレイリストからサムネイル画像を取得するのかを選択させ、その後、選択されたプレイリストの再生を開始する。そして、ステップS25において、ユーザにサムネイルにしたいシーンを選択させる。

【0108】ステップS23、あるいは、ステップS25において、サムネイル画像が選択されると、ステップS26に進む。ステップS26以降の処理は、図30に示したフローチャートのステップS5以降の処理と同様であるので、その説明は省略する。

【0109】このように、サムネイル画像をプレイリストに付与することにより、それらのサムネイル画像を管理しやすくなり、記録されているAVストリームを再生する際にも、再生させるAVストリームの選択などの処理を簡便に行なわせることが可能となる。

【0110】図32は、ディスク上につくられるディレクトリおよびファイルの別の例を示す図である。“menu.tidx”と“menu.tdat”は、メニューサムネール、すなわちVolumeを代表する1つのピクチャおよびPlayList毎に、それを代表する1つのピクチャの情報をストアする。すべてのメニューサムネールのヘッダ情報は、1つのmenu.tidxに集めて管理される。すべてのメニューサムネールのピクチャデータは、1つのmenu.tdatに集められて管理される。

【0111】“mark.tidx”と“mark.tdat”は、マークサムネール、すなわちマーク点で指されるピクチャについての情報をストアする。Volume中のすべてのClipおよびPlayListに付加されている、全てのマークサムネールのヘッダ情報は、1つのmark.tidxに集めて管理される。全てのマークサムネールのピクチャデータは、1つのmark.tdatに集めて管理される。

【0112】すなわち、このファイル構造では、サムネールのヘッダ情報とサムネールのピクチャデータが、別

々のファイルに分けて管理される。この理由は、ファイルシステムが行うファイルの管理方法に関係する。すなわち、サムネールのヘッダ情報のファイルは、比較的重要な情報であるため、ファイルシステムがディスク上にデータを2重書きして管理する。

【0113】これは、例えば、片方のデータがディスクの傷などによって失われてしまったような場合に対応できるようにするためであり、データのバックアップの意味合いをもつ。ピクチャデータのファイルは、2重書きはしないが、その理由は、ピクチャデータは比較的データ量が大きくなり、これを2重書きするとディスク上の必要容量が大きくなるためである。

【0114】menu.tdatおよびmark.tdatにストアされるピクチャデータの符号化方法、サンプリング構造、およびスキャン構造は、ピクチャ毎に符号化方法を変更することが可能であるが、全てのピクチャデータで同じであるほうが、記録再生装置1の構成を簡単化できる。例えば、JFIF (JPEG File Interchange Format)、コンポーネント信号、プログレッシブスキャンフォーマットが使用される。

【0115】これら4個のファイルのシンタクスとセマンティクスを説明する。“menu.tidx”と“mark.tidx”は、同じシンタクス構造を持つ。図33は、“menu.tidx”と“mark.tidx”的シンタクス構造を示す。version_numberは、このサムネールヘッダ情報ファイルのバージョンナンバーを示す4個の数字である。

【0116】lengthは、このlengthフィールドの直後のバイトからmenu.tidx/mark.tidxの最後のバイトまでのバイト数である。number_of_thumbnailsは、menu.tidxの場合にはmenu.tdatにストアされているサムネールピクチャの数であり、mark.tidxの場合、mark.tdatにストアされているサムネールピクチャの数である。

【0117】tn_block_sizeは、menu.tidxの場合、menu.tdatの中の1つのtn_blockのサイズを示し、mark.tidxの場合、mark.tdatの中の1つのtn_blockのサイズを示す。このサイズは、1024バイトを単位とする大きさである。例えば、tn_block_size=1は、1つのtn_blockのサイズが1024バイトであることを示す。1つのサムネールピクチャは、1つのtn_blockの中にストアされなければならない。

【0118】number_of_tn_blocksは、menu.tidxの場合、menu.tdatの中にあるtn_blockの数を示し、mark.tidxの場合にはmark.tdatの中にあるtn_blockの数を示す。thumbnail_indexは、このthumbnail_indexフィールドに続くサムネール情報のインデックス番号を表す。thumbnail_indexして、0xFFFFという値を使用してはならない。

【0119】menu.tidxの場合、thumbnail_indexはUIAppInfoVolume()、UIAppInfoPlayList()の中のref_thumbnail_indexによって参照される。mark.tidxの場合、thum

bnail_indexはPlayListMark()およびClipMark()の中のref_thumbnail_indexによって参照される。

【0120】ref_to_tn_block_idは、menu.tidxの場合、menu.tdat中の1つのtn_blockを示し、そのtn_blockは、thumbnail_indexで指されるピクチャデータをストアしている。ref_to_tn_block_idの値は、menu.tdatのシンタクス中のtn_block_idの値を参照する。

【0121】mark.tidxの場合、mark.tdat中の1つのtn_blockを示し、そのtn_blockは、thumbnail_indexで指されるピクチャデータをストアしている。ref_to_tn_block_idの値は、menu.tdatのシンタクス中のtn_block_idの値を参照する。

【0122】picture_byte_sizeは、thumbnail_indexで指される1つの符号化サムネールピクチャのデータ長をバイト単位で示す。picture_byte_sizeは、1024*tn_block_sizeの値以下でなければならない。すなわち、記録再生装置1は、1つの符号化サムネールピクチャのデータ長を1024*tn_block_sizeの値以下になるように、符号化しなければならない。

【0123】horizontal_picture_sizeは、thumbnail_indexで指される符号化サムネールピクチャの水平方向の画素数を示す。vertical_picture_sizeは、thumbnail_indexで指される符号化サムネールピクチャの垂直方向の画素数を示す。display_aspect_ratioは、thumbnail_indexで指される符号化サムネールピクチャのディスプレイ・アスペクト・レシオを示す。値の意味を図34に示す。

【0124】color_spaceは、Y,Cb,Crのコンポーネント信号をR,C,Bのコンポーネント信号へ変換する時のフォーマットを示す。値の意味を図35に示す。

【0125】"menu.tdat"と"mark.tdat"は、同じシンタクス構造を持つ。図36は、"menu.tdat"と"mark.tdat"のシンタクス構造を示す図である。tn_blockは、1つの符号化サムネールピクチャがストアされる領域である。1つのサムネールピクチャのバイト長は、1つのtn_blockの大きさ以下である。1つのピクチャデータの第1バイト目は、tn_blockの第1バイト目と一致していなければならない。

【0126】menu.tdatの場合、1つのtn_blockのサイズは、menu.tdatの中のtn_block_sizeで示される。mark.tdatの場合、1つのtn_blockのサイズはmark.tdatの中のtn_block_sizeで示される。各tn_blockは、それがシンタクス中のfor-loopの中で現れるときのtn_block_idの値で区別される。menu.tidx中のtn_block_idは、menu.tidx中のref_to_tn_block_idによって参照される。mark.tidx中のtn_block_idは、mark.tidx中のref_to_tn_block_idによって参照される。

【0127】サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は、容易に高速に実行できなければならない。この理由のため、menu.tdatとmar

k.tdatは、ブロック構造を有する。1つのピクチャデータは、1つのtn_blockに格納される。

【0128】menu.tdatおよびmark.tdatのtn_block列の中に、使用されていないtn_blockが存在してもよい。例えば、所定のサムネイルの削除をする場合、サムネールのヘッダ情報ファイルの中にエントリーされているthumbnail_indexを消去し、サムネールのピクチャデータファイルを何も変更しなかったとき、tn_block列の中に、使用されていないtn_blockができる。

【0129】図37は、サムネールピクチャデータが、どのようにtn_blockに格納されるかを模式的に表した図である。図37に示すように、1つのサムネイルピクチャのバイト長は、1つのtn_blockの大きさ以下である。tn_block列の中に、使用されていないtn_blockが存在してもよい。

【0130】上述した実施の形態においては、例えば、menu.tbatとmark.tbatのファイルが2個に分けられて記録されるとして説明したが、それぞれのファイルをブロックと考え、menu.tbatのデータの第1のブロックと、mark.tbatのデータの第2のブロックを、1つのファイルにまとめて記録するようにしても良い。1つのファイルにまとめて記録するとは、例えば、Clip Information fileの中で、SequenceInfo、CPI、ClipMarkを別のブロックとして記録するような形態にしても良い。

【0131】上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば図38に示すような構成の汎用のパーソナルコンピュータなどに、記録媒体からインストールされる。

【0132】図38に示すパーソナルコンピュータにおいて、CPU(Central Processing Unit)201は、ROM(Read Only Memory)202に記憶されているプログラム、または記憶部208からRAM(Random Access Memory)203にロードされたプログラムに従って各種の処理を実行する。RAM203にはまた、CPU201が各種の処理を実行する上において必要なデータなども適宜記憶される。

【0133】CPU201、ROM202、およびRAM203は、バス204を介して相互に接続されている。このバス204にはまた、入出力インターフェース205も接続されている。

【0134】入出力インターフェース205には、キーボード、マウスなどよりなる入力部206、CRT、LCDなどよりなるディスプレイ、並びにスピーカなどよりなる出力部207、ハードディスクなどより構成される記憶部208、モ뎀、ターミナルアダプタなどより構成され

る通信部209が接続されている。通信部209は、ネットワークを介しての通信処理を行う。

【0135】入出力インターフェース205にはまた、必要に応じてドライブ210が接続され、磁気ディスク221、光ディスク222、光磁気ディスク223、或いは半導体メモリ224などが適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部208にインストールされる。

【0136】この記録媒体は、図38に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク221（フロッピディスクを含む）、光ディスク222（CD-ROM（Compact Disk-Read Only Memory）、DVD（Digital Versatile Disk）を含む）、光磁気ディスク223（MD（Mini-Disk）を含む）、若しくは半導体メモリ224などよりなるパッケージメディアにより構成されるだけでなく、コンピュータに予め組み込まれた状態でユーザに提供される、プログラムが記憶されているROM202や記憶部208が含まれるハードディスクなどで構成される。

【0137】なお、本明細書において、媒体により提供されるプログラムを記述するステップは、記載された順序に従って、時系列的に行われる処理は勿論、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

【0138】また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

【0139】

【発明の効果】以上の如く、本発明の記録装置および方法、並びに第1のプログラムにおいては、入力された動画像データから、当該動画像データを代表するサムネイル画像を第1のサムネイルデータとして生成すると共に、動画像データから、特徴的な画像として抽出された画像のサムネイル画像か、または、ユーザが指定した画像のサムネイル画像を第2のサムネイルデータとして生成し、第1のサムネイルデータと第2のサムネイルデータを、各々独立したグループとして記録媒体に記録するようにしたので、その記録されている画像データを用いることにより、ユーザが記録されているデータの選択を簡便に行える。

【0140】また、本発明の再生装置および方法、並びに第2のプログラムによれば、AVストリームの再生が指示された場合、AVストリームの内容を示すサムネイル画像の画像データを管理する管理データを読み出し、読み出し手段により読み出された管理データに基づき、画像データを読み出し、AVストリームから、特徴的な画面として抽出された画面のサムネイル画像の画像データか、または、ユーザにより指定された画面のサムネイル画像の画像データと、その画像データを管理する管理データ

を読み出し、読み出された管理データに基づき、画像データを読み出すようにしたので、その読み出された管理データにより、ユーザが記録されているデータの選択を簡便に行える。

【図面の簡単な説明】

【図1】本発明を適用した記録再生装置の一実施の形態の構成を示す図である。

【図2】ClipとPlaylistの関係を表す図である。

【図3】AVストリームを管理する構造のUML図である。

【図4】DVRシステムのディレクトリ構成を表す図である。

【図5】info.dvrを説明する図である。

【図6】DVRVolume()を説明する図である。

【図7】UIAppInfoVolume()を説明する図である。

【図8】TableOfPlayLists()を説明する図である。

【図9.】UIAppInfoPlaylist()を説明する図である。

【図10】zzzz.clpiを説明する図である。

【図11】ClipMark()を説明する図である。

【図12】マークを説明する図である。

【図13】xxxx.rpls、yyyy.vplsを説明する図である。

【図14】Playlist()を説明する図である。

【図15】PlayItem()を説明する図である。

【図16】PlayListMark()を説明する図である。

【図17】メニュー・サムネイルを説明する図である。

【図18】プレイリストに付けられるマークを説明する図である。

【図19】クリップに付けられるマークを説明する図である。

【図20】サムネイルを格納するファイルを説明する図である。

【図21】menu_thmb/mark.thmbを説明する図である。

【図22】Thumbnail()を説明する図である。

【図23】Thumbnail_picture_formatを説明する図である。

【図24】サムネイルの画像データをtn_block()に格納する方法を説明する図である。

【図25】他のディレクトリ・ファイル構造を示す図である。

【図26】図25に示したファイル構造に対応するmenu_thmb/mark.thmbを説明する図である。

【図27】図25に示したファイル構造に対応するTableOfPlayLists()を説明する図である。

【図28】図25に示したファイル構造に対応するPlaylist()を説明する図である。

【図29】図25に示したファイル構造に対応するUIAppInfoPlaylist()を説明する図である。

【図30】マーク・サムネイルの作成手順を表したフローチャートである。

【図31】メニュー・サムネイルの作成手順を表したフロ

ーチャートである。

【図3.2】DVRシステムの他のディレクトリ構成を表す図である。

【図3.3】サムネールのヘッダ情報ファイルのシンタクスを示す図である。

【図3.4】display_aspect_ratioを説明する図である。

【図3.5】color spaceを説明する図である。

【図3.6】サムネールのピクチャデータファイルのシルエットを示す図である。

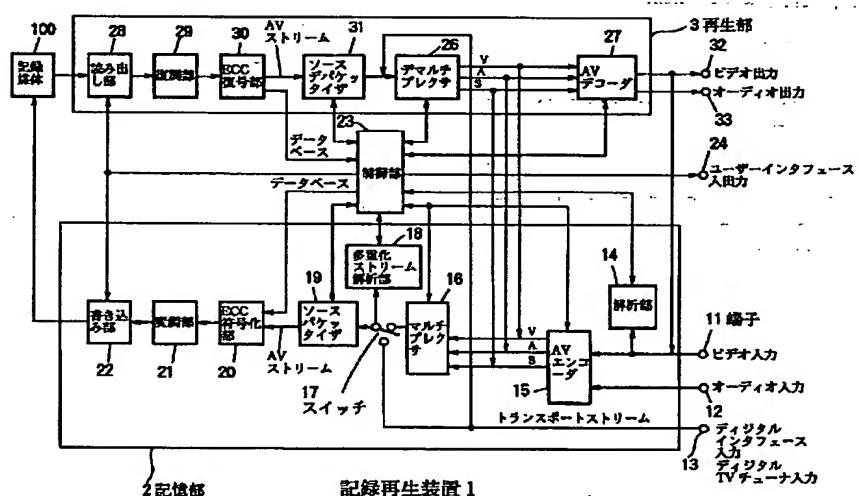
【図3.7】 tn_blockへのデータの格納について説明する図である。

【図3.8】媒体を説明する図である。

【符号の説明】

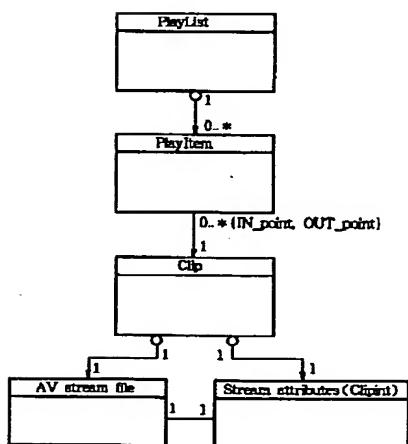
1 記録再生装置, 11乃至13 端子, 14 解析部, 15 AVエンコーダ, 16 マルチブレクサ, 17 スイッチ, 18 多重化ストリーム解析部, 19 ソースパケッタイザ, 20 ECC符号化部, 21 変調部, 22 書き込み部, 23 制御部, 24 ユーザインタフェース, 25 スイッチ, 26 デマルチブレクサ, 27 AVデコーダ, 28 読み出し部, 29 復調部, 30 ECC復号部, 31 ソースパケッタイザ, 32, 33 端子

[圖 1]



[図3]

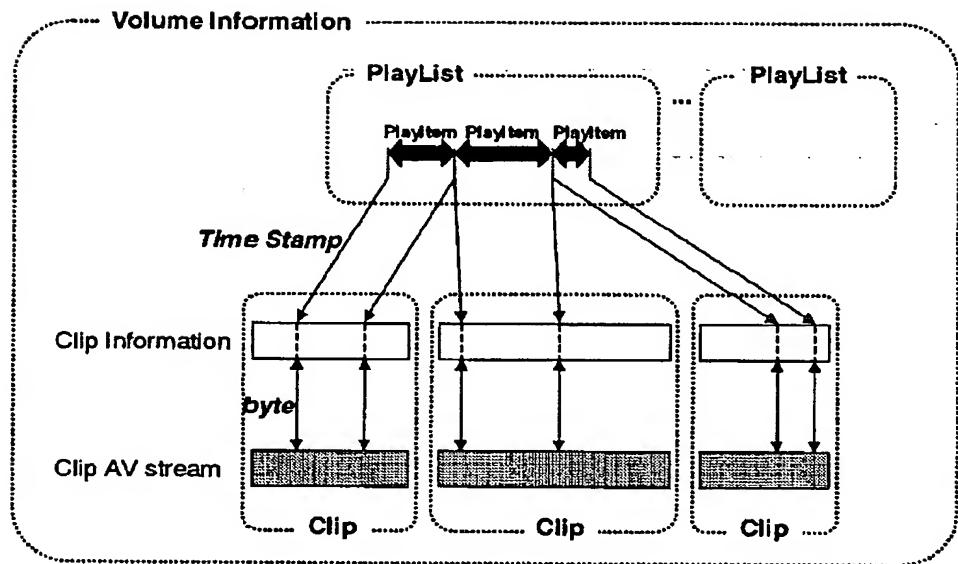
[図6]



Syntax	No. of bits	Mnemonics
DVRVolume()		
<u>version_number</u>	8*4	bsbf
length	32	uimsbl
ResumeVolume()		
UIApplInfoVolume()		

DVR Volume のシンタクス

【図2】

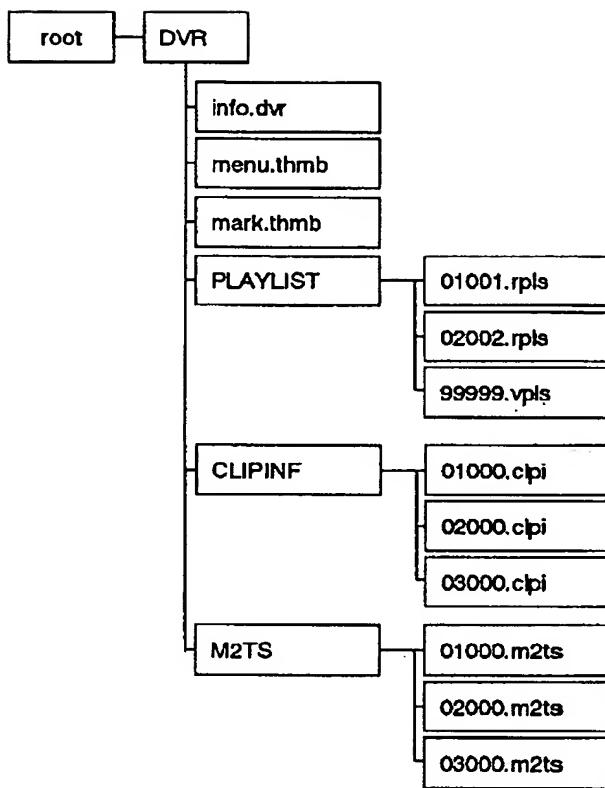


【図5】

Syntax	No. of bits	Mnemonics
info.dvr {		
<u>TableOfPlayLists_Start_address</u>	32	uimsbf
<u>MakerPrivateData_Start_address</u>	32	uimsbf
reserved	192	bslbf
<u>DVRVolume()</u>		
for(i=0; i<N1; i++){		
padding_word	16	bslbf
}		
<u>TableOfPlayLists()</u>		
for(i=0; i<N2; i++){		
padding_word	16	bslbf
}		
<u>MakerPrivateData()</u>		
}		

info.drv のシンタクス

【図4】



【図7】

Syntax	No. of bits	Mnemonics
UIAppInfoVolume 0 {		
character_set	8	bslbf
name_length	8	uimsbf
Volume_name	8*256	bslbf
reserved	15	bslbf
Volume_protect_flag	1	bslbf
PIN	8*4	bslbf
ref_thumbnail_index	16	uimsbf
reserved	7	bslbf
rp_info_valid_flag	1	uimsbf
rp_ref_to_PlayList_file_name	8*10	bslbf
rp_ref_to_PlayItem_id	16	uimsbf
rp_time_stamp	32	uimsbf
}		

【図 8】

■ TableOfPlayLists - シンタクス (4.2.3.2 の別案)

Syntax	No. of bits	Mnemonics
TableOfPlayLists0 {		
version number	8*4	bslbf
length	32	uimbsbf
number of PlayLists	16	uimbsbf
for (i=0; i<number of PlayLists; i++) {		
PlayList file name	8*10	bslbf
UIAppInfoPlayList 2()		
}		
}		

TableOfPlayLists のシンタクス

【図9】

Syntax	No. of bits	Mnemonics
UIAppInfoPlayList20 {		
character_set	8	bslbf
name_length	8	uimsbf
PlayList_name	8*256	bslbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
valid_period	4*8	bslbf
maker_id	16	bslbf
maker_code	16	bslbf
reserved	11	bslbf
playback_control_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
archive	2	bslbf
ref_thumbnail_index	16	uimsbf
reserved	7	bslbf
rp_info_valid_flag	1	uimsbf
rp_ref_to_PlayItem_id	16	uimsbf
rp_time_stamp	32	uimsbf
reserved_for_future_use	240	bslbf
}		

【図11】

Syntax	No. of bits	Mnemonics
ClipMark() {		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for(i=0; i < number_of_Clip_marks; i++) {		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	bslbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

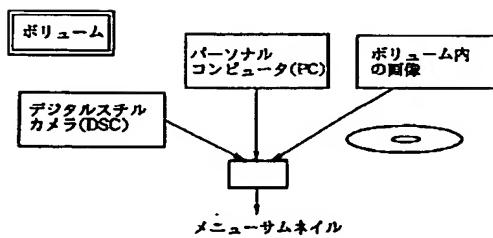
ClipMark のシンタクス

【図10】

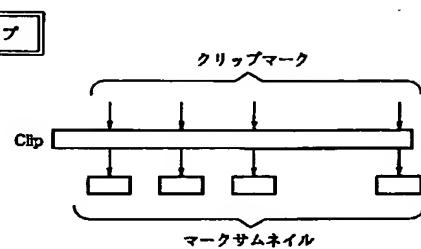
Syntax	No. of bits	Mnemonics
<code>zzzzz.cipi {</code>		
<code>STC Info Start address</code>	32	uimsbf
<code>ProgramInfo Start address</code>	32	uimsbf
<code>CPI Start address</code>	32	uimsbf
<code>ClipMark Start address</code>	32	uimsbf
<code>MarkerPrivateData Start address</code>	32	uimsbf
<code>reserved</code>	96	bslbf
<code>ClipInfo()</code>		
<code>for(i=0; i<N1; i++) {</code>		
<code>padding word</code>	16	bslbf
<code>}</code>		
<code>STC Info()</code>		
<code>for(i=0; i<N2; i++) {</code>		
<code>padding word</code>	16	bslbf
<code>}</code>		
<code>ProgramInfo()</code>		
<code>for(i=0; i<N3; i++) {</code>		
<code>padding word</code>	16	bslbf
<code>}</code>		
<code>CPI()</code>		
<code>for(i=0; i<N4; i++) {</code>		
<code>padding word</code>	16	bslbf
<code>}</code>		
<code>ClipMark()</code>		
<code>for(i=0; i<N5; i++) {</code>		
<code>padding word</code>	16	bslbf
<code>}</code>		
<code>MarkerPrivateData()</code>		
<code>}</code>		

zzzzz.cipi のシンタクス

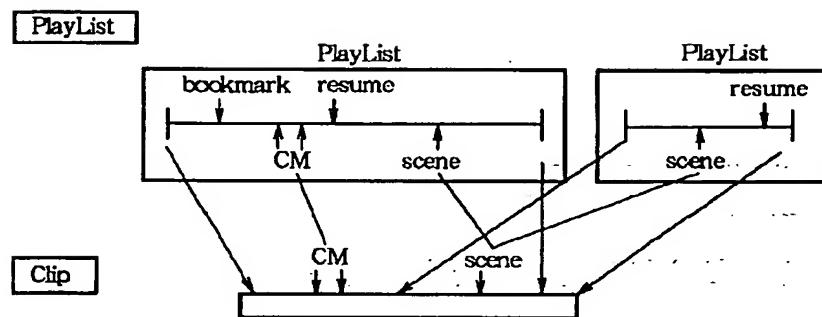
【図17】



【図19】



【図12】



PlayList 上のマークと Clip 上のマーク

【図13】

Syntax	No. of bits	Mnemonics
xxxxx.rpls / yyyy.ypls {		
PlayListMark Start address	32	uimbsbf
MakerPrivateData Start address	32	uimbsbf
reserved	192	bslbf
PlayList()		
for(i=0; i<N1; i++){		
padding word	16	bslbf
}		
PlayListMark()		
for(i=0; i<N2; i++){		
padding word	16	bslbf
}		
MakerPrivateData()		
}		

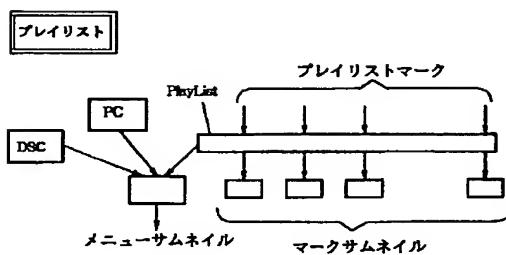
xxxxx.rpls と yyyy.ypls のシナジー

【図14】

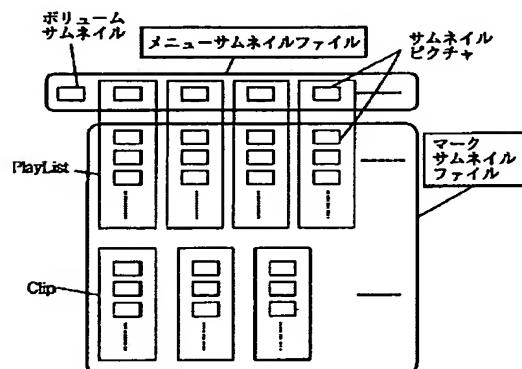
Syntax	No. of bits	Mnemonics
PlayList()		
version_number	8*4	bslbf
length	32	uimsbf
PlayList_type	8	uimsbf
CPI_type	1	bslbf
reserved	7	bslbf
UIAppInfoPlayList()		
number_of_PlayItems // main path	16	uimsbf
if (<Virtual PlayList>){		
number_of_SubPlayItems // sub path	16	uimsbf
}else{		
reserved	16	bslbf
}		
for (PlayItem_Id=0;		
PlayItem_id < number_of_PlayItems;		
PlayItem_id++) {		
PlayItem() // main path		
}		
if (<Virtual PlayList>){		
if (CPI_type==0 && PlayList_type==0){		
for (i = 0; i < number_of_SubPlayItems; i++)		
SubPlayItem() // sub path		
}		
}		
}		

PlayList のシンタクス

【図18】



【図20】



【図15】

Syntax	No. of bits	Mnemonics
PlayItem() {		
Clip_information_file_name	8*10	bslbf
reserved	24	bslbf
STC_sequence_id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
reserved	14	bslbf
connection_condition	2	bslbf
if (<Virtual Playlist>) {		
if (connection_condition=='10') {		
BridgeSequenceInfo()		
}		
}		
}		

PlayItem のシンタクス

【図23】

Thumbnail_picture_format	Meaning
0x00	MPEG-2 Video I-picture
0x01	DCF (restricted JPEG)
0x02	PNG
0x03-0xff	reserved

thumbnail_picture_format

【図27】

Syntax	No. of bits	Mnemonics
TableOfPlayLists() {		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++) {		
PlayList_file_name	8*10	bslbf
}		
}		

TableOfPlayLists のシンタクス

【図16】

Syntax	No. of bits	Mnemonics
PlayListMark()		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayList_marks	16	ulmsbf
for(i=0; i < number_of_PlayList_marks; i++) {		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
PlayItem_Id	8	uimsbf
reserved	24	uimsbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		

PlayListMark のシナリオ

【図 21】

Syntax	No. of bits	Mnemonics
menu.thumb / mark.thumb {		
reserved	256	bslbf
Thumbnail()		
for(i=0; i<N; i++)		
padding_word	16	bslbf
}		
}		

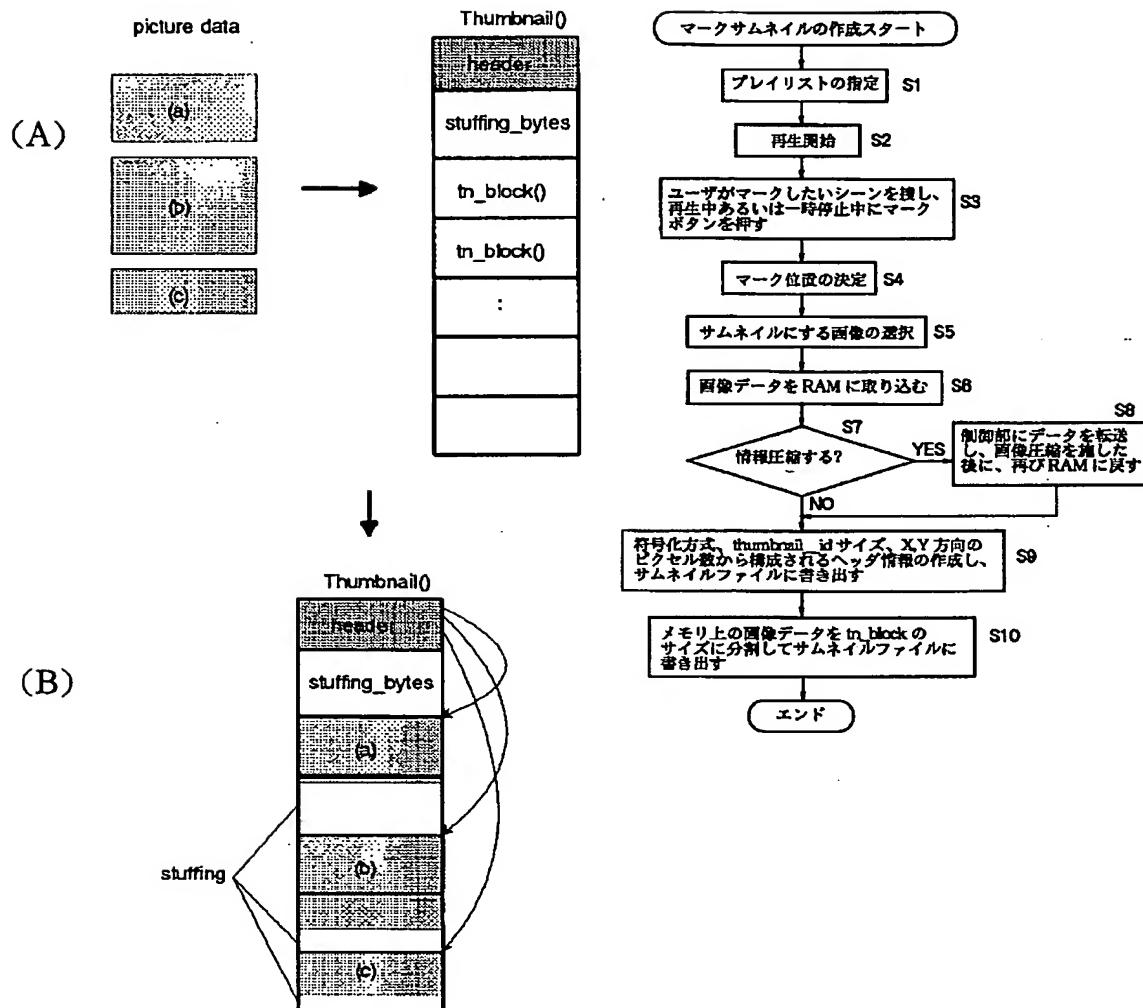
menu.thumb と mark.thumb のシンタクス

【图22】

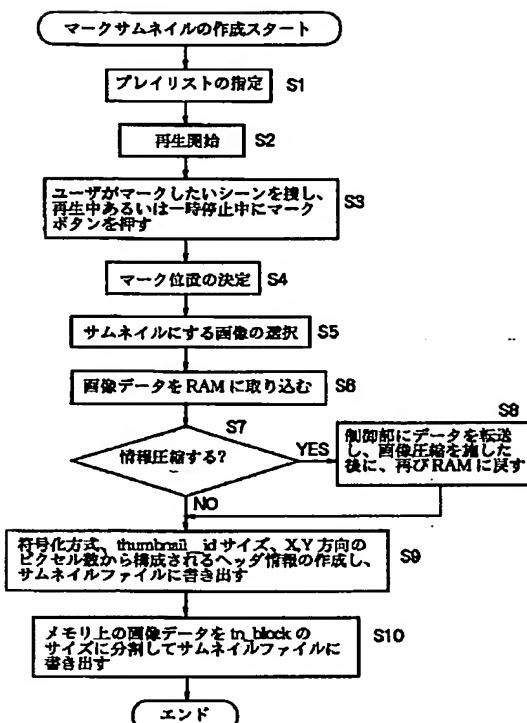
Syntax	Bits	Mnemonics
Thumbnail()		
version number	8*4	char
length	32	uimsbf
if (length != 0) {		
tn blocks start address	32	bslbf
number of thumbnails	16	uimsbf
tn block size	16	uimsbf
number of tn blocks	16	uimsbf
reserved	16	bslbf
for(i = 0; i < number of thumbnails; i++) {		
thumbnail index	16	uimsbf
thumbnail picture format	8	bslbf
reserved	8	bslbf
picture data size	32	uimsbf
start tn block number	16	uimsbf
x picture length	16	uimsbf
y picture length	16	uimsbf
reserved	16	uimsbf
}		
stuffing bytes	8*2^L1	bslbf
for(k = 0; k < number of tn blocks; k++) {		
tn_block	fixed	

Thumbnail のシンタクス

【図 24】



【図 30.】

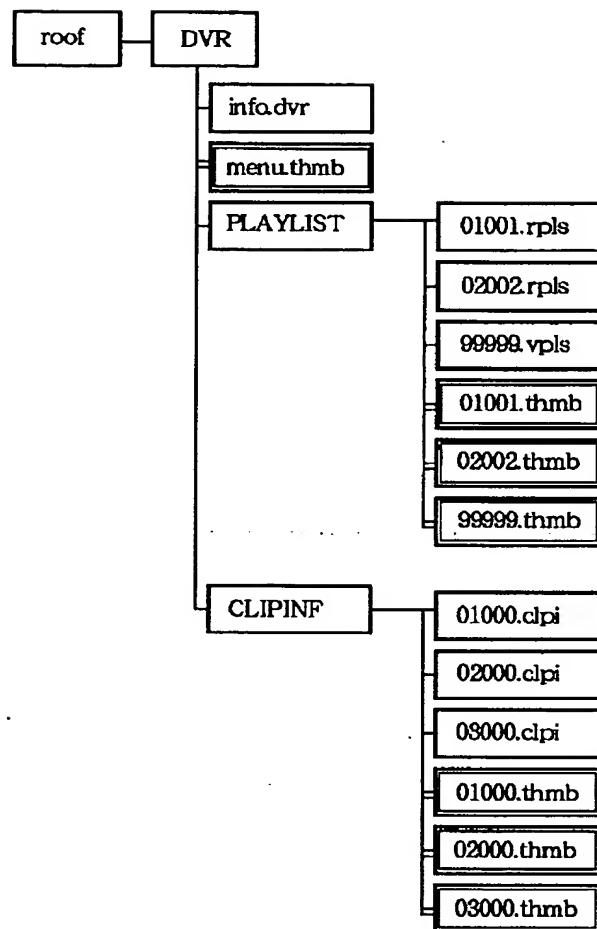


【図 34】

display_aspect_ratio

display_aspect_ratio	Meaning
0, 1	reserved for future use
2	4:3 display aspect ratio
3	16:9 display aspect ratio
4-15	reserved for future use

【図 25】



【図 35】

color_space

display_aspect_ratio	Meaning
0	BT 709
1	ITU-R Rec.601
2-15	reserved for future use

【図26】

Syntax	No. of bits	Mnemonics
menu.thumb / xxxx thumb {		
reserved	256	bslbf
Thumbnail()		
for(i=0; i<N; i++)		
padding_word	16	bslbf
}		

menu.thumb と xxxx thumb のシンタクス

【図28】

Syntax	No. of bits	Mnemonics
PlayList()		
version_number	8*4	bslbf
length	32	uimsbf
PlayList_type	8	uimsbf
CPI_type	1	bslbf
reserved	7	bslbf
UIAppInfoPlayList0		
number_of_PlayItems	// main path	uimsbf
if (<Virtual PlayList>){		
number_of_SubPlayItems	// sub path	uimsbf
}else{		
reserved	16	bslbf
}		
for (PlayItem_id=0;		
PlayItem_id < number_of_PlayItems;		
PlayItem_id++) {		
PlayItem0	// main path	
}		
if (<Virtual PlayList>){		
if (CPI_type==0 && PlayList_type==0){		
for (i = 0; i < number_of_SubPlayItems; i++)		
SubPlayItem0	// sub path	
}		
}		

PlayList のシンタクス

【図29】

Syntax	No. of bits	Mnemonics
UIAppInfoPlayList()		
character_set	8	bslbf
name_length	8	uimsbf
PlayList_name	8*256	bslbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
valid_period	4*8	bslbf
maker_id	16	uimsbf
maker_code	16	uimsbf
reserved	11	bslbf
playback_control_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
archive	2	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	240	bslbf
}		

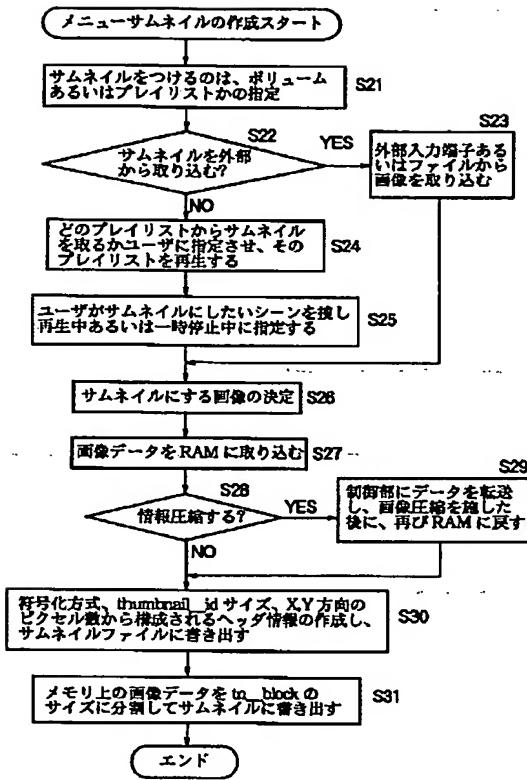
UIAppInfoPlayList のシンタクス

【図33】

Syntax	No. of bits	Mnemonic
menu.tidx / mark.tidx {		
version_number	8*4	char
reserved_for_future_use	256	bslbf
length	32	uimebf
if (length != 0) {		
number_of_thumbnails	16	uimsbf
tn_block_size	16	uimsbf
number_of_tn_blocks	16	uimsbf
for(i=0; i<number_of_thumbnails; i++) {		
thumbnail_index	16	uimsbf
ref_to_tn_block_id	16	uimsbf
picture_byte_size	32	uimsbf
horizontal_picture_size	16	uimsbf
vertical_picture_size	16	uimsbf
display_aspect_ratio	4	uimesbf
color_space	4	uimsbf
reserved_for_word_align	8	bslbf
}		
}		
}		

サムネールのヘッダ情報ファイルのシンタクス

【図 3 1】

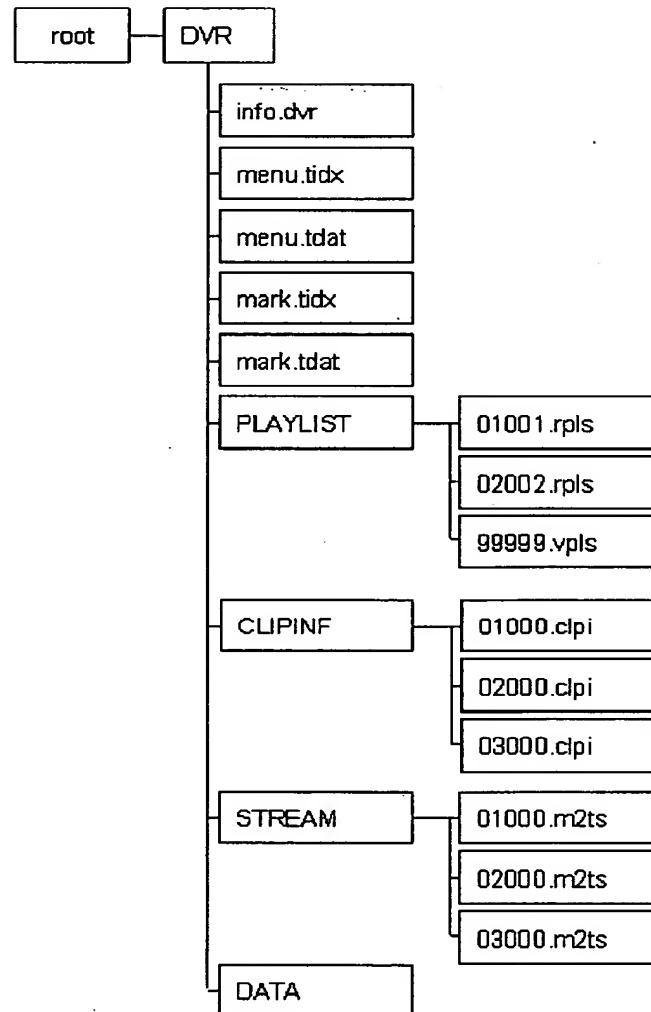


【図 3 6】

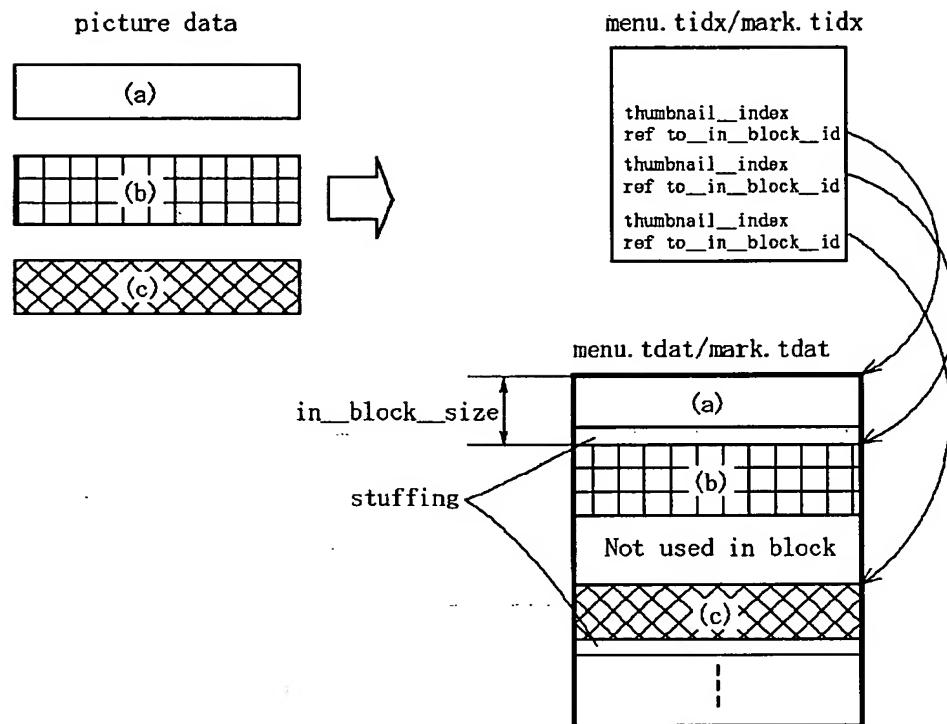
Syntax	No. of bits	Mnemonic
menu.tdat / mark.tdat { for(<i>tn_block_id</i> =0; <i>tn_block_id</i> < <i>number_of_tn_blocks</i> ; <i>tn_block_id</i> ++) { <i>tn_block</i> } } }		
	<i>tn_block_size</i> *1024* 8	

サムネールのピクチャデータファイルのシンタクス

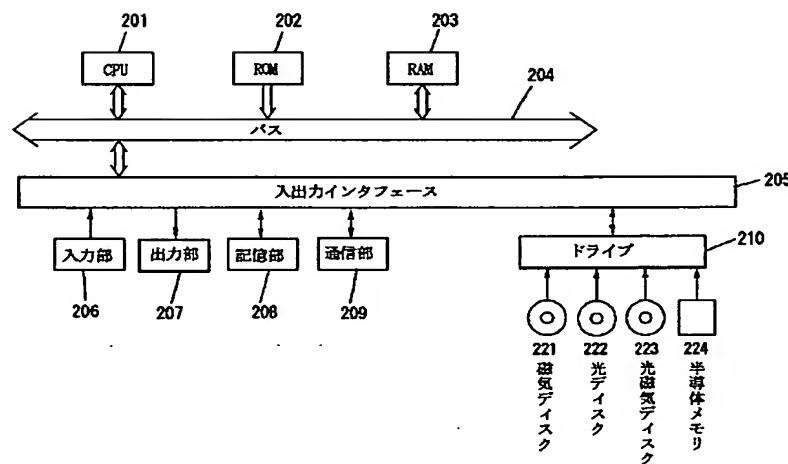
【図3 2】



【図37】



【図38】



フロントページの続き

F ターム(参考) 5C052 AA02 AC08 CC11 DD10
5C053 FA14 FA20 FA23 GB06 GB09
GB38 HA30 KA01 KA08 KA24
KA26 LA06 LA07
5D044 AB05 AB07 BC04 CC06 DE22
DE38 DE53 EF05 FG18
5D110 AA17 AA19 AA27 AA29 DA01
DA06 DA11 DA20 DB03 DC05
DC15 FA08 FA09